Metodologia d'un anàlisi de malware

Memòria final del projecte

Autor: Guillem Gordillo Garcia

Director: Manel Medina Llinàs

Data defensa: 23/04/18

Centre: FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Universitat: UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) – BarcelonaTech

RESUM

Aquest projecte explica una metodologia per a analitzar si un programa és maliciós fent servir els tipus d'anàlisi estàtic i dinàmic, juntament amb la contextualització de l'increment de programes maliciosos i els diferents tipus. Durant l'anàlisi, s'obté informació respecte el programa analitzat que s'anirà fent servir, mitjançant diferents eines, per comprovar si es tracta de malware.

RESUMEN

Este proyecto explica una metodología para analizar si un programa es malicioso usando los tipos de análisis estático y dinámico, junto con la contextualización del incremento de programes maliciosos y los diferentes tipos. Durante el análisis, se obtiene información respecto el programa analizado que se irá usando, mediante diferentes herramientas, para comprobar si se trata de malware.

ABSTRACT

This project explains a methodology to analyze whether a program is malicious using the static and dynamic analysis, together with the contextualization of the increase of malware and the different kinds. During the analysis, we will obtain information regarding the analyzed program which be used, with different tools, in order to conclude its maliciousness.

ÍNDEX

ÍNDEX DE FIGURES

ÍNDEX DE TAULES

INTRODUCCIÓ

Actualment hi ha moltes tecnologies de seguretat que intenten evitar que hi hagi forats a les xarxes que intentem protegir, però hi ha molt poques tecnologies que siguin capaces d'entendre el malware que ha infectat aquesta xarxa. La majoria d'aquestes tecnologies no serveixen en el cas d'una infecció, i és quan entendre i saber l'objectiu del programari maliciós és important per poder mitigar i resoldre el problema, així com prevenir que torni a passar.

Què és el malware?

Considerem malware tot aquell software que tingui per objectiu comprometre un usuari, ordinador o xarxa indiferentment de la finalitat. Existeixen diferents tipus de malware, entre ells hi ha: virus, troyans, worms, rootkits, scareware, spyware, adware, etc. Es cataloga cadascun d'ells per característiques com el funcionament i l'objectiu. [1]

Què és l'anàlisi de malware?

Entenem l'anàlisi de malware com a l'acció d'inspeccionar el software maliciós amb l'objectiu d'identificar-ho, entendre el comportament i el funcionament, i buscar la manera d'eliminar-ho i protegir-se'n.

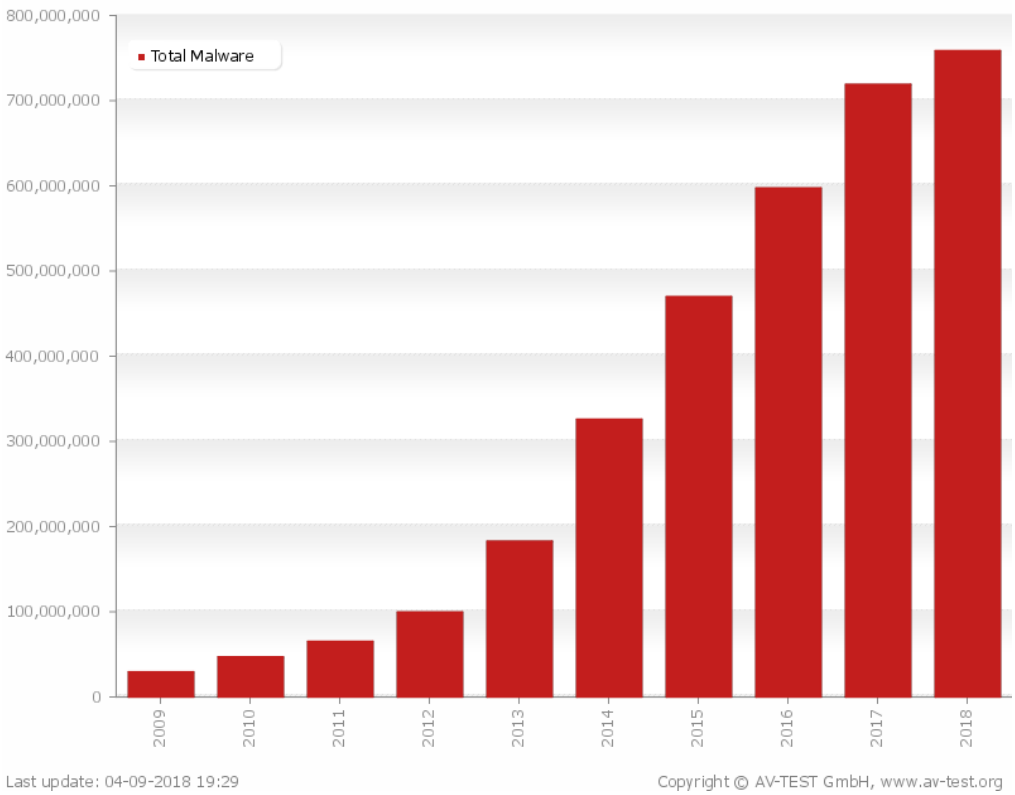
ABAST

En aquest projecte, es realitzarà un anàlisi de malware amb la finalitat de poder dir amb seguretat si un programa és maliciós o no fent servir la metodologia indicada. Per a aconseguir aquesta fita es faran servir un conjunt d'eines públiques per tothom i es documentarà quins passos mínims seguir per a arribar a una conclusió respecte el subjecte.

CONTEXT

Existeixen milions de programes maliciosos per internet, i dia rere dia n'hi ha més, però la majoria d'empreses i organitzacions només tenen un antivirus per a comprovar que un programa descarregat no és maliciós. A més a més, hi ha moltes tecnologies per evitar que s'infectin aquestes xarxes, tals com un Firewall, Firewall d'Aplicacions Web (WAF), Intrusion Detection Prevention System (IDPS), etc.

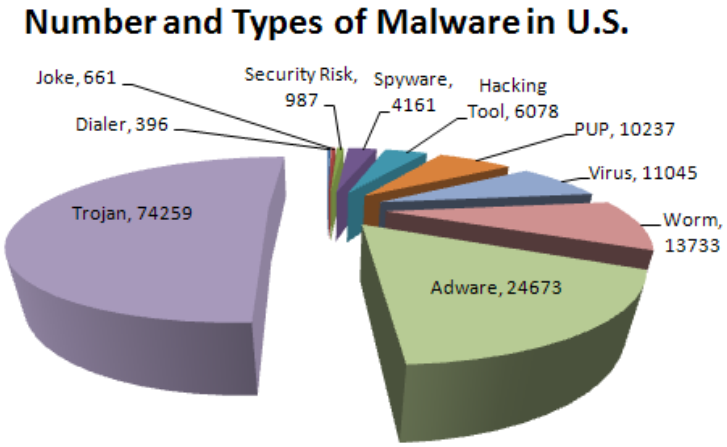
De totes aquestes, només l'IDPS seria capaç d'evitar una infecció dins la xarxa interna un cop es coneixen les signatures que fa servir el malware o el comportament del mateix. Què passa si és un malware nou del qual se'n desconeix el comportament i la signatura? És per això que l'anàlisi de malware és tant important per entendre i protegir del programari maliciós.



Il·lustració 1 estadístiques de l'increment de malware per AV-test [3]

A la il·lustració anterior es pot veure que la quantitat de malware conegut no ha parat de créixer des del 2012. S'ha de tenir en compte que la gràfica anterior no conta dues vegades un malware que ja havia aparegut, per tant tot el malware és nou.

De tot aquest malware que apareix a la gràfica, es cataloguen en diferents subtipus com es pot veure a la il·lustració següent:



Il·lustració 2 subtipus de malware

La il·lustració 2 deixa evident que el Troyà és el tipus de malware més extès que es coneix actualment. A més a més, l'adware té una forta presència entre els programes maliciosos i seguidament hi ha els cucs i els virus. Més endavant explicaré cada tipus de malware més detalladament per poder valorar mentre fem un anàlisi contra què ens enfrontem.

Per aturar aquest software maliciós existeixen principalment els antivirus, que treballen de dues maneres: tracten de comparar signatures de cada programa fent servir un algoritme determinat, i comprovar que no encaixin amb cap signatura que tinguin a la base de dades. Aquesta base de dades conté la gran majoria de signatures que s'obtenen un cop s'ha arribat a la conclusió que està relacionat amb activitats sospitoses, en funció de cada antivirus. L'altra manera de funcionament és: mitjançant heurístiques busquen patrons de conducta per descobrir malware desconegut fins ara, comparen comportaments amb malware conegut i en funció de la similitud cataloguen un executable com a maliciós. Aquesta última tecnologia és la coneguda amb el nom de "Endpoint Protection", tot i que està comprovat que és factible saltar-se aquesta protecció si el malware carrega les seves funcionalitats a memòria en temps d'execució.

Per estudiar els diferents antivirus que existeixen actualment al mercat una font a tenir en compte és el Gartner Magic Quadrant d'antivirus. Aquest quadrant cataloga entre l'habilitat d'executar les funcionalitats que proporcionen i com de completa és la visió que tenen de la seguretat.

Il·lustració 3 Gartner Magic Quadrant per Endpoint Protection

A la il·lustració 3 podem veure un conjunt d'empreses que ofereixen serveis de Endpoint Protection. Cal destacar que les empreses en millor posició són Trend Micro, Sophos, Kaspersky Lab i Symantec.

Aquestes empreses ofereixen serveis molt similars a l'anàlisi de malware, executant de manera automatitzada un fitxer que creiem que és maliciós i oferint dades sobre el comportament del mateix. Aquesta execució es fa a màquines virtuals o "sandboxes" que contenen un sistema operatiu determinat i un programari limitat que explicaré en l'apartat d'anàlisi dinàmic. A més a més, tot i que no està verificat ja que no donen informació sobre com treballa el producte, després de llegir molt sobre els productes sembla que realitzen un tipus d'anàlisi sense executar el malware per treure una primera conclusió sobre el programa.

Moltes vegades els programes maliciosos s'executen només si tenim programes de tercers instal·lats com per exemple el Microsoft Word a una versió en concret, o el Adobe Reader. Llavors, es pot donar el cas que el programa maliciós aparenti ser legítim i que

n'obtinguem una conclusió errònia ja que no s'ha executat la part maliciosa que porta, com per exemple una macro del Microsoft Word.

Per tant, en la metodologia que seguirem es farà servir màquines virtuals amb diferent programari, a vegades des-actualitzat, i diverses eines amb diferents funcionalitats que s'explicaran més endavant a la metodologia.

És necessari per a prosseguir amb l'anàlisi de malware, tenir coneixements del tipus de programari maliciós que ens podem trobar [2]. Tot seguit hi ha una llista dels més comuns:

Backdoor
Codi maliciós que s'instal·la a un dispositiu per permetre l'accés a un atacant. Els backdoors normalment permeten que l'atacant es connectin amb poca o gens autenticació i poden executar comandes en el sistema local.

Botnet
Actua de manera similar a un backdoor però tots els dispositius infectats amb la mateixa botnet reben instruccions d'un servidor Command-And-Control (C&C) per realitzar accions.

Downloader
Un downloader és codi maliciós que descarrega un altre codi maliciós. Els downloaders normalment són instal·lats en el sistema de manera silenciosa quan aconsegueixen accés al sistema. El downloader descarregarà tot el codi necessari per acabar de penetrar el sistema.

Information-stealing malware
Malware que recull informació de la seva víctima i normalment l'envia a l'atacant. Exemples d'aquest tipus de malware són: sniffers, password hash grabbers i keyloggers.

Launcher
Un Launcher, de manera similar al downloader, serveix per executar altres programes maliciosos. Normalment els launchers fan servir tècniques no tradicionals per executar aquests programes amb l'objectiu d'aconseguir menys visibilitat o més accessos del sistema.

Worm

Un cuc és un programa que es replica a si mateix i destrueix les dades i fitxers de l'ordinador. El cuc, treballa per menjar-se els fitxers del sistema operatiu i les dades relacionades fins que el disc és buit.

Virus

De manera similar, un virus s'intenta propagar després de adaptar-se a un altre tros de codi per reproduir-se quan s'executa. La majoria de les vegades és reprodueix per sistemes de fitxers compartits.

Adware

La intenció d'un adware és la de mostrar o descarregar publicitat en el dispositiu en que s'ha instal·lat.

Trojan

El més conegut i més perillós malware és el Troyà. L'objectiu d'aquest malware són les dades fiscals de la víctima o qualsevol altre tipus d'informació que pugui permetre aconseguir aquest objectiu.

Rootkit

Un rootkit és el malware més difícil d'exterminar i més letal per als dispositius de les víctimes ja que permet l'obtenció d'informació a altres tipus de malware així com possibles facilitats d'intrusió. Un rootkit sol elevar privilegis al màxim i s'amaga de manera molt persistent al dispositiu. Es recomana esborrar el disc dur sencer un cop infectat per aquest tipus de malware.

Ransomware

El tipus de malware que més noticies ha creat últimament és el Ransomware. Aquest malware intenta encriptar el disc dur de la víctima i demanar un "ransom" o rescat a canvi de la clau per desencriptar el disc. [4]

Rogue software i scareware

Es tracta d'un programa que intenta enganyar o estafar als usuaris per pretendre donar una solució a un altre malware trobat dins del sistema i fer que es descarregui aquest programa maliciós. Un cop descarregat, sol fer creure a l'usuari que ha netejat el dispositiu de malware però en veritat està introduint-se ell mateix de manera persistent.

Cal remarcar que entre els analistes de malware, els intercanvis d'informació el realitzen a partir de indicadors de compromís o IOC. Un IOC és una dada observada a la xarxa o en un sistema operatiu que ens permet dir amb certesa que hi ha hagut una intrusió. Els típics IOCs que es fan servir són signatures de fitxers, adreces IP, dominis relacionats amb activitats sospitoses, etc.

És important tenir present que s'ha de comparar la informació que obtenim durant l'anàlisi per veure si la podem relacionar amb algun IOC que ens permeti concloure que es tracta de malware.

OBJECTIUS

Aquest projecte se centra en realitzar anàlisis de malware així com preparar l'entorn per això, donat l'increment de les infeccions en el temps i l'impacte que aquestes provoquen. A més a més, dona valor a diferents eines Open Source i demostrar les seves funcionalitats.

Per tant, d'objectius específics tenim els següents:

- Preparar un entorn segur minimitzant els riscos associats a una possible infecció per malware.
- Demostrar la utilitat dels productes Open Source d'aquest projecte.
- Posar en valor els programes Open Source.
- Donar unes directrius sobre com realitzar un anàlisi de malware.
- Fer diferents proves d'anàlisi per contrastar la metodologia.
- Conscienciació dels possibles objectius de programes maliciosos.
- Fer accessible la metodologia a persones mínimament tècniques per a, en cas de necessitat, analitzar un programa per verificar si té un objectiu maliciós.

METODOLOGIA

La finalitat d'aquest projecte és concloure una metodologia a seguir en el cas que ens trobem amb una possible mostra de malware. És per això, que s'aniran fent proves amb diferents eines en cada fase de l'anàlisi de malware per arribar a la conclusió de si es tracta de malware o no.

S'ha de tenir en compte l'entorn que servirà de laboratori per a fer els anàlisis de malware. Com bé he comentat, aquest entorn ha de ser virtual i ha de tenir un conjunt de software típic que el malware sol explotar per aconseguir el seu objectiu.

És per això, que es crearà una màquina virtual fent servir VirtualBox, ja que dona la possibilitat de fer "snapshots" de l'estat de la màquina en tot moment. Com que l'entorn de laboratori està pensat per a fer molts anàlisis de malware, és important poder revertir l'estat de la màquina virtual a com es trobava just abans de fer l'anàlisi, ja que pot haver estat infectat i per tant ens estalviaria el temps de tornar a crear l'entorn de zero fent servir els "snapshots" o imatges. Una imatge d'una màquina virtual és un fitxer que es pot carregar amb totes les configuracions i aplicacions instal·lades que en el mateix moment en que es va realitzar la imatge. Aquestes imatges són gestionades des del mateix programa de Virtualbox.

Virtualbox [5] és programari lliure desenvolupat per l'empresa Oracle Corporation. Permet l'ús de màquines virtuals i té suport per moltes plataformes. Per al nostre laboratori haurem de fer servir una màquina virtual per fer l'anàlisi estàtic.

Aquesta màquina virtual disposarà de Windows 10 instal·lat, ja que és el sistema operatiu més estès i amb més vulnerabilitats conegudes després de Windows 7. Cal tenir en compte que s'ha d'assemblar a un entorn de treball d'una persona dins d'una empresa per a aparentar una situació real.
La metodologia d'aquest projecte es dividirà en els següents apartats:

- Preparació de l'entorn
- Static Analysis
- Dynamic Analysis

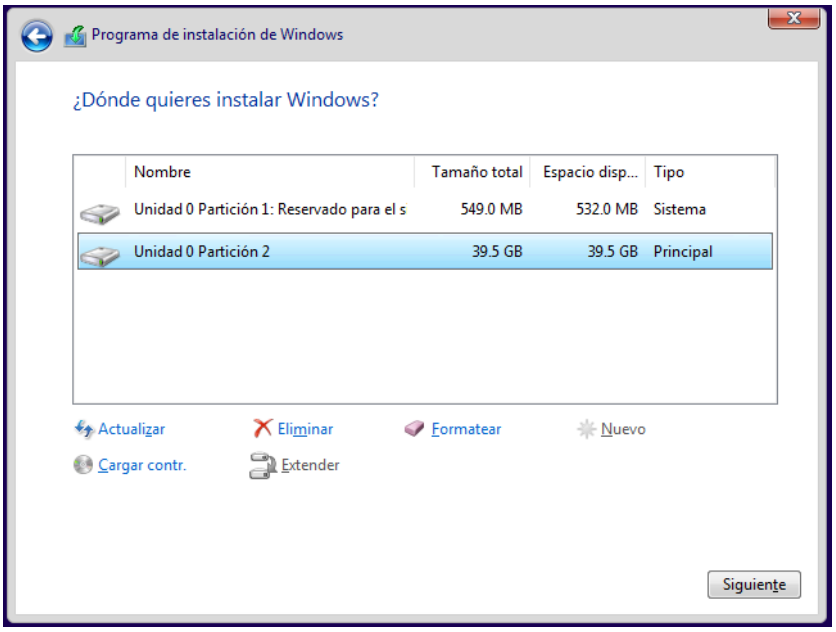
PREPARACIÓ DE L'ENTORN

Com bé hem comentat, per al nostre entorn necessitarem d'un ordinador amb capacitat suficient per mantenir dues màquines virtuals actives.

La primera amb Windows 10 i la segona amb qualsevol altra distribució que pugui fer servir els programes de l'anàlisi dinàmic. En el meu cas, he fet servir ubuntu [6] ja que es tracta d'un sistema operatiu més lleuger que Windows i molt fàcil de fer servir.

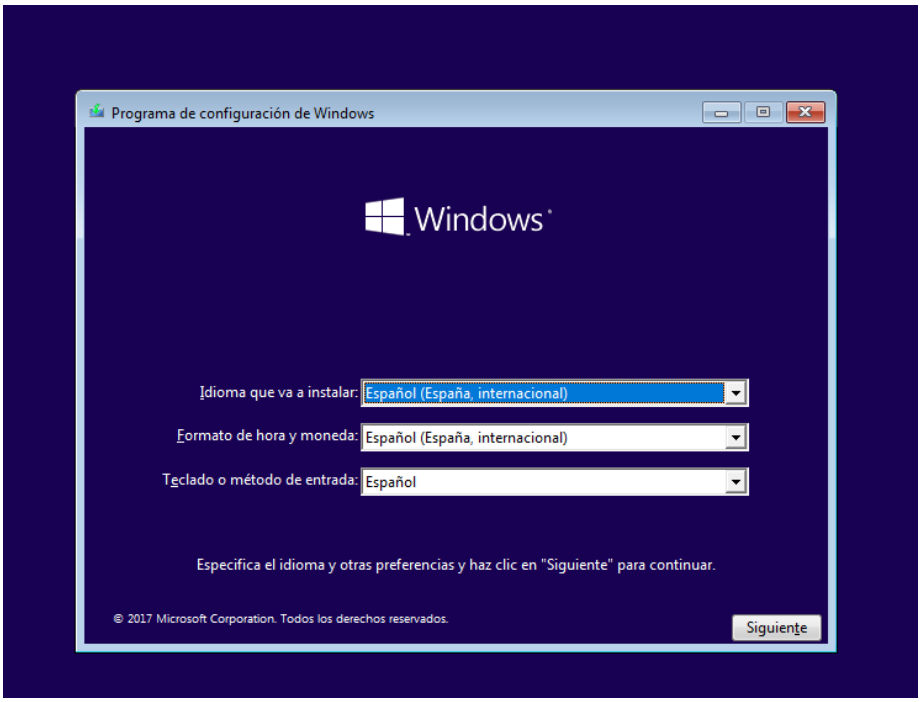
Primer de tot, haurem de crear les noves màquines virtuals un cop ens descarreguem les imatges d'internet. En aquesta metodologia no es profunditzarà en el procés de creació de les màquines virtuals, però és donarà una imatge general dels passos a seguir.

Al crear la primera màquina virtual que serà la del sistema operatiu Windows, li he donat 40 GB d'espai de disc com es pot veure a la següent imatge.



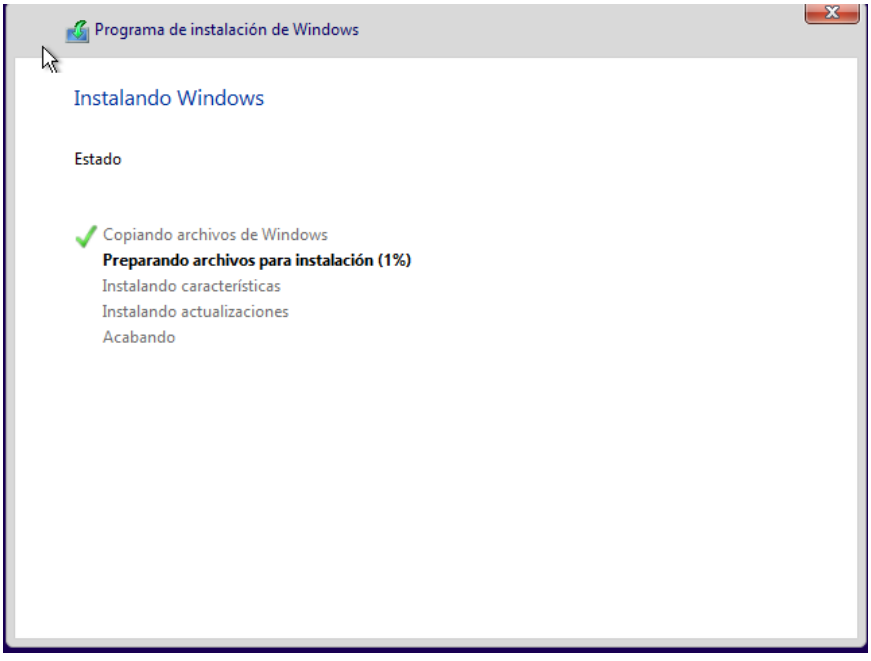
Il·lustració 4 Particions windows.

Seguidament, simplement cal adjuntar el fitxer "iso" del Windows 10 i seguir amb el procés d'instal·lació d'un sistema operatiu normal.



Il·lustració 5 Inici instal·lació Windows

Al omplir totes les dades necessàries, arribarà un moment en que l'últim pas de la instal·lació apareix, com es pot veure a la següent imatge:



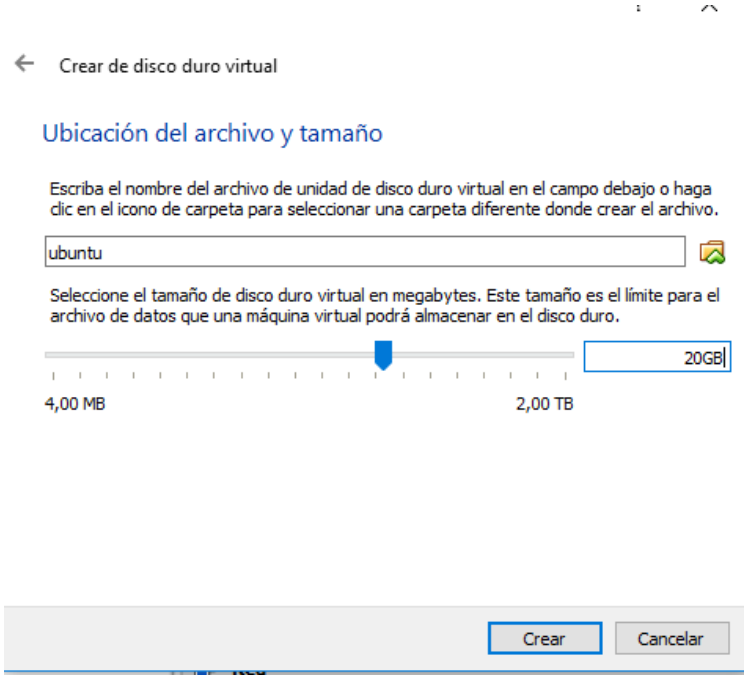
Il·lustració 6 Instal·lació Windows.

Un cop acaba el procés de la imatge anterior, ja tenim el sistema operatiu instal·lat. És recomanable afegir ara el programari que farem servir durant els anàlisis de malware. Entre ells, és recomanable instal·lar els següents a la màquina Windows:

- Procmon [7]
- Regshot [8]
- Strings [9]
- Wireshark [12]
- PeView [13]
- Resource Hacker
- Process Explorer
- MD5deep

Aquests programes es poden trobar als enllaços de l'apartar de bibliografia i són molt senzills d'instal·lar.

Seguidament, fem el mateix procés però ara amb la màquina que tindrà per sistema operatiu Ubuntu. Donat que el sistema operatiu d'Ubuntu és més lleuger que el de Windows, l'espai que necessita és molt menor i per això li he posat 20 GB com es pot veure a la imatge següent:



Il·lustració 7 Configuració disc Ubuntu.

Un cop s'ha afegit la nova màquina virtual i s'ha assignat el fitxer "iso" que instal·larà el sistema, s'ha de seguir el procediment que apareix en pantalla. A la següent imatge es mostra l'inici d'una instal·lació del sistema operatiu Ubuntu.

Il·lustració 8 Instal·lació Ubuntu.

Com abans, es recomana instal·lar els següents programes a la màquina virtual amb sistema operatiu Ubuntu per tal d'estar preparat per a la realització de l'anàlisi de malware. Podeu trobar enllaços als programes a l'apartat de bibliografia.

- INetSim
- ApateDNS

Un cop tenim les dues màquines virtuals connectades entre elles per a que hi hagi comunicació, podem procedir a l'anàlisi de malware.

ANÀLISIS ESTÀTIC

Aquest anàlisi es basarà principalment en obtenir informació del programa de manera fàcil i ràpida per a fer una primera teoria sobre el subjecte que analitzarem. Ens valdrem de fonts externes per comprovar si aquest programa ja ha sigut analitzat com per exemple VirusTotal. A més a més, observarem les dependències entre fitxers i les llibreries que importa, així com la mida de les dades del programa.

Aquest tipus d'anàlisi és efectiu contra malware poc desenvolupat o ja conegut, però si el programa maliciós és sofisticat ens pot enganyar i fer pensar que es tracta de programari lícit i és per això que no és concloent.

Per a desenvolupar aquesta part de l'anàlisi farem servir les següents eines, que s'explicaran al moment de fer-les servir:

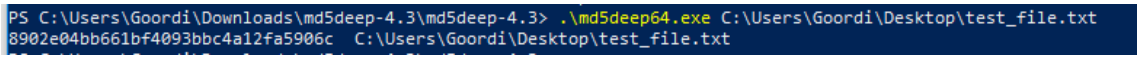
- MD5deep
- Radare2 o strings
- PEview
- Resource Hacker

Verificació de fitxers

Primer de tot cal comparar aquest fitxer amb d'altres per veure si ja existeixen informes sobre el comportament d'aquest. Per a realitzar la comparació es podria anar comparant bit a bit tots els fitxers, però donat que és molt costós el que se sol fer servir és el *hashing*. El *hashing* serveix per identificar un fitxer de qualsevol mida en un conjunt de caràcters hexadecimal de una mida fixa normalment més petita.

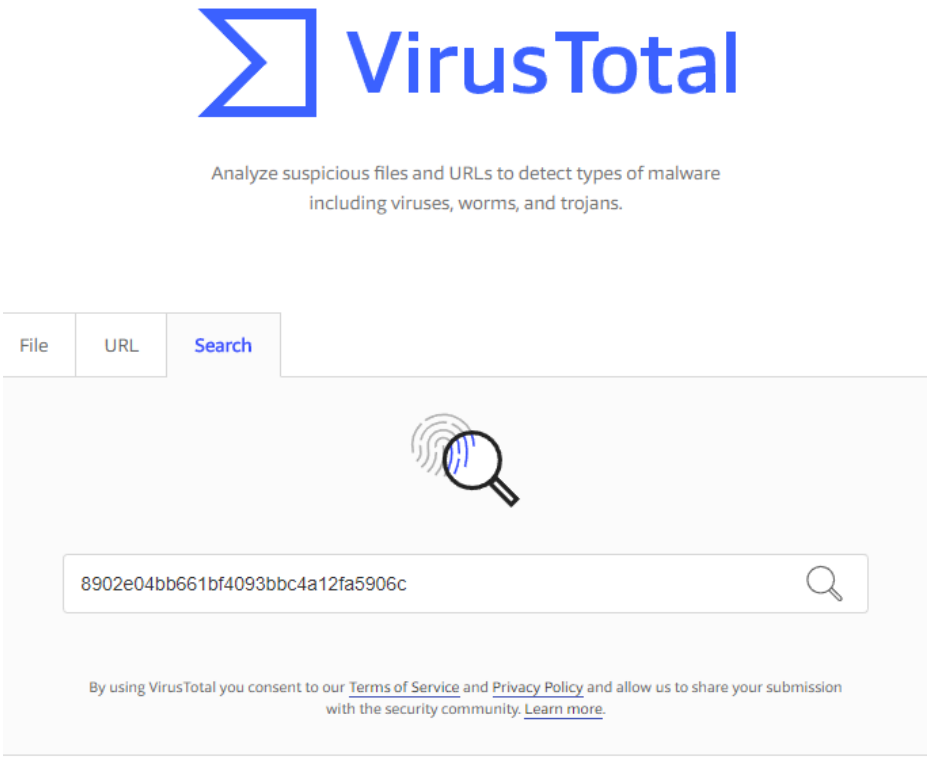
Per obtenir el *hash* d'un fitxer el que farem serà aplicar l'algoritme MD5 (Message Digest Algorithm 5) fent servir l'eina MD5deep**.** Aquesta eina obté de manera ràpida una seqüència de 32 dígits hexadecimals que identificarà el fitxer. Tot i que no permet reconstruir les dades originals, podem comparar amb bases de dades de malware conegut per fer un primer filtre sobre la intenció del fitxer.

Com a bases de dades farem servir VirusTotal, que compara un conjunt de Antivirus de *hashes* i, a més, moltes vegades trobem un primer anàlisi de comportament que veurem més endavant.



Il·lustració 9 Exemple md5deep.

Amb aquest hash md5 podem accedir a la pàgina web de virustotal i fer una cerca com es pot veure a la imatge següent.



Il·lustració 10 Portal VirusTotal.

Un cop li donem a cercar, ens redirigeix a una pàgina web que ens informa sobre tota la informació que disposa d'aquest fitxer. Altrament, podem accedir al següent enllaç que és exactament on ens ha redirigit un cop li hem donat a cercar.

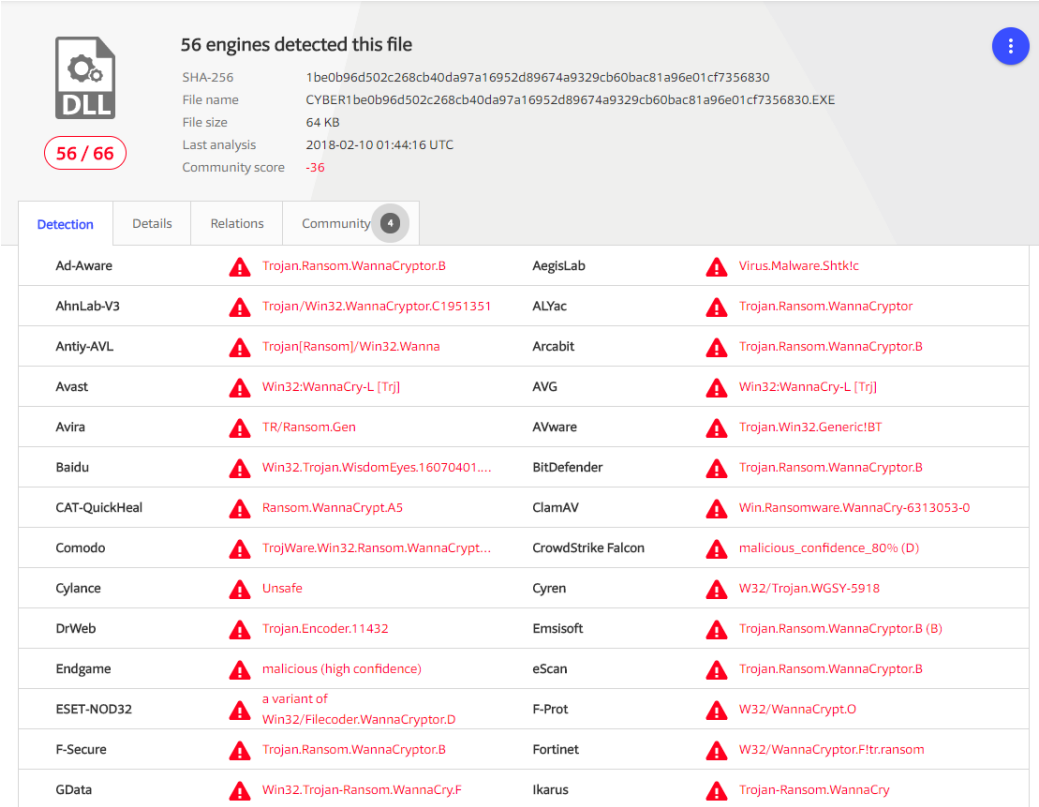
https://www.virustotal.com/#/search/8902e04bb661bf4093bbc4a12fa5906c

A l'enllaç anterior, ja que era una prova, ens diu que el fitxer no s'ha trobat a la seva base de dades i per tant no té informació.

Si fem la cerca sobre un malware conegut en canvi, ens retorna la quantitat d'antivirus que l'han catalogat com a maliciós. Per a fer una prova hem fet servir el següent enllaç:

https://www.virustotal.com/#/file/1be0b96d502c268cb40da97a16952d89674a9329cb60bac81a96e01cf7356830/detection

El md5 que hem buscat pertany al malware conegut com a WannaCry, i hi ha certs antivirus que el cataloguen directament per donar tota la informació del mateix. A la següent imatge tenim les deteccions del WannaCry.



Il·lustració 11 Exemple malware a VirusTotal.

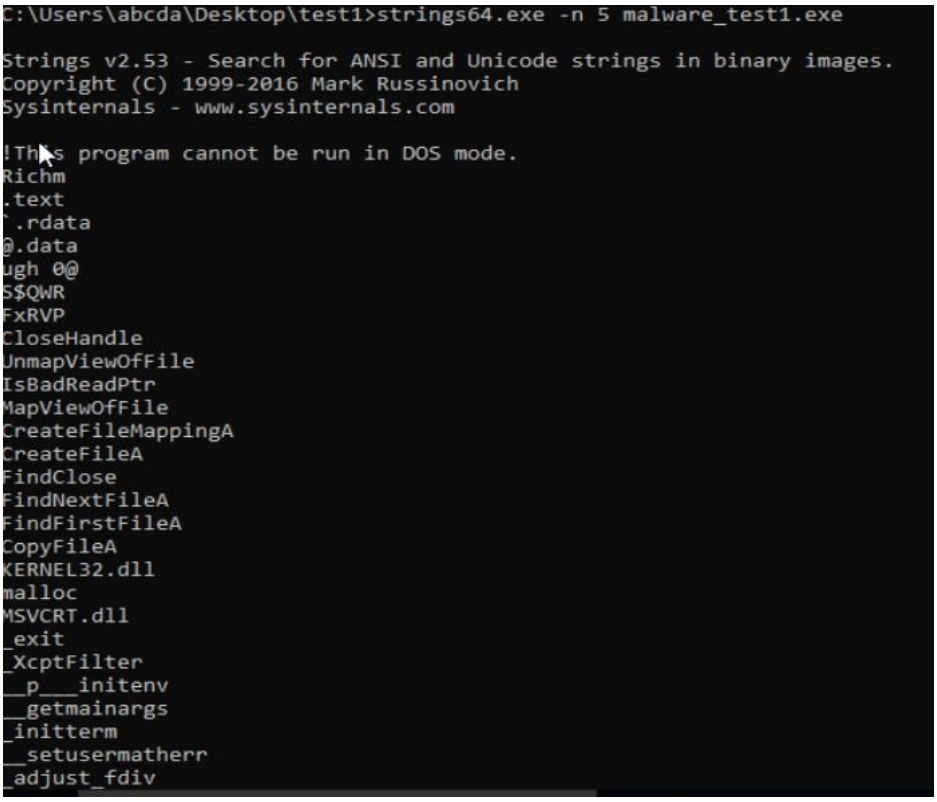
És important fer servir tota la informació que ens ofereix VirusTotal i per tant, a l'apartat "Details", tenim indicadors del comportament d'aquest programa com per exemple el comportament a la xarxa i molt important és el que ha comentat la comunitat respecte aquest. Moltes vegades una altra persona ja s'ha vist infectada i ho deixa per escrit a internet.

Tot i això, com he comentat abans, simplement al realitzar qualsevol modificació sobre aquest fitxer el hash obtingut al aplicar l'algoritme MD5 ja no és el mateix, i és per això que aquest resultat anterior no és concloent en cas que no s'hagi trobat cap coincidència o el nombre d'antivirus que l'ha detectat és baix.

Strings

Un cop fet un primer filtre de malware conegut, és interessant investigar sobre les cadenes de caràcters d'una longitud més gran o igual que tres, o altrament coneguts com a strings. Hi ha un programa anomenat "strings" que la seva funció principal és aquesta.

A la següent il·lustració es pot veure un exemple d'execució del programa strings amb l'argument "-n" que indica el nombre de caràcters en ASCII necessaris per treure-ho per pantalla. La comanda usada és *#strings64.exe -n 5 malware_test1.exe.*



Il·lustració 12 Exemple execució strings

Aquest programa, té diverses vulnerabilitats que poden fer que un malware s'executi de manera no desitjada. En funció dels paràmetres que hem fet servir pot ser que ens infecti la màquina on estem fent les proves.

Per el motiu anterior, personalment, prefereixo fer servir radare2 per a aquesta fita tot i que no és l'única manera. Radare2 és un projecte open source que va començar com a una eina de forense per a editar a partir de scripts de línia de comandes l'hexadecimal de fitxers. Posteriorment es van afegir funcionalitats que l'han decantat cap al "reversing", que tracta d'entendre i ser capaç de reproduir el malware des de zero.

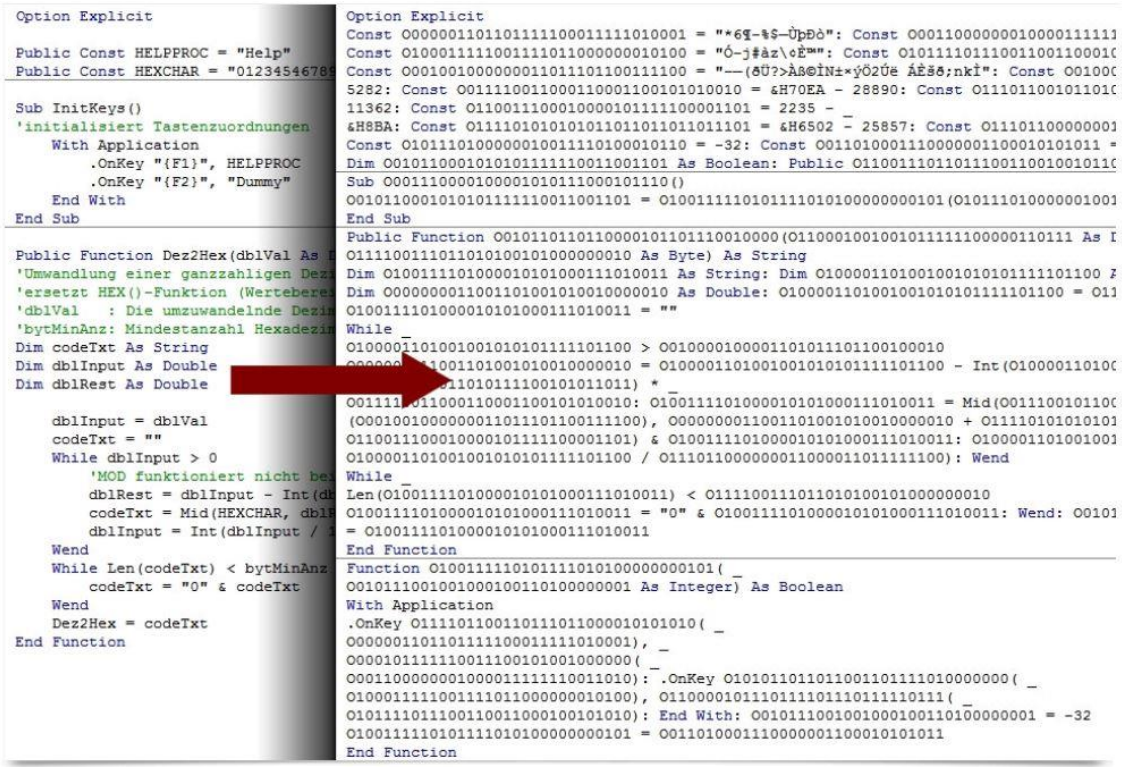
Un exemple d'execució de radare2 per obtenir els strings del fitxer */bin/ls* seria la següent: *#r2 -AA /bin/ls .*

Indiferentment del programa fet servir, hem obtingut els strings del codi maliciós que ens oferiran molta informació que anirem cercant posteriorment durant els diferents tipus d'anàlisi. Sobretot cal ser perspicaç i saber aplicar les dades que tenim. Un exemple seria cercar un domini o una IP obtinguda abans, a les connexions que es realitzen a nivell de xarxa per veure quina és la funció del domini o IP.

Moltes vegades les persones que desenvolupen malware intenten evitar que els responsables de Seguretat puguin esbrinar què ha estat en execució en el seu sistema

ja sigui perquè han desplegat algun tipus de porta del darrere que no volen que es descobreixi o perquè volen replicar l'atac posteriorment. És una manera de dificultar també la descoberta de la vulnerabilitat que ataquen per a que no pugui ser arreglada a la següent actualització, si és que no ho està.

Per a amagar el malware el que se sol fer és ofuscar o empaquetar el codi. Ofuscar és l'acte d'amagar el codi mitjançant altre codi que no deixa clar què s'està executant o que simplement el que fa és afegir soroll a l'execució. Un programa empaquetat són un subconjunt de programes en que el codi maliciós és comprimit i no pot ser analitzat.



Il·lustració 13 Exemple codi ofuscat

A la imatge anterior es pot veure un exemple de codi ofuscat en que s'han creat nous bucles i s'ha generat nou codi simplement per fer que el comportament i el codi del programa no sigui trivial de descobrir.

En el cas que el programa estigui empaquetat, no podrem veure tots els strings fins que no es carreguin en memòria per executar-se. Per determinar si un programa està empaquetat una manera ràpida i força eficaç és contant el nombre de cadenes que ens han aparegut. En termes generals, si n'hi ha molt poques podem concloure que està empaquetat, però si n'hi ha força sol ser que no ho està o almenys no en la seva totalitat.

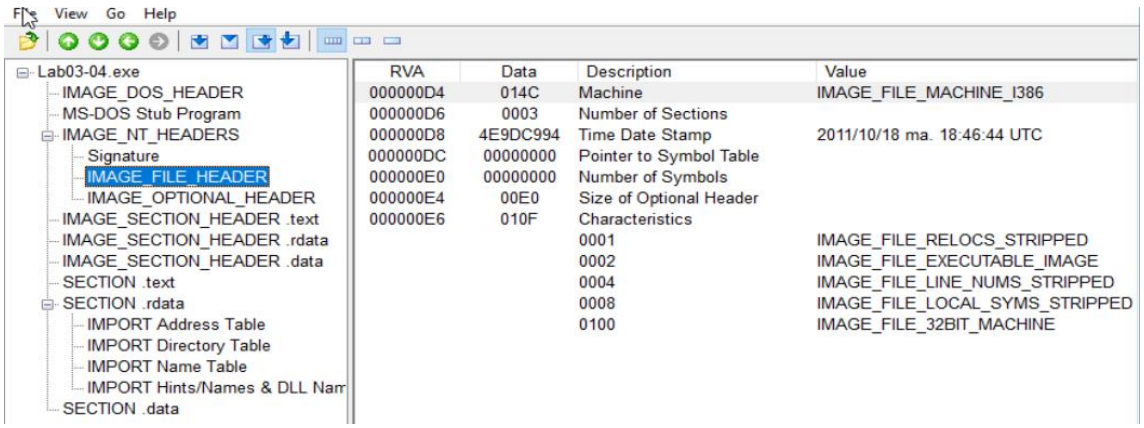
Portable executable

Portable executable format (PE) és el format usat per la majoria d'executables de Windows i llibreries.

Les capçaleres dels fitxers en format PE poden oferir molta més informació de la que hem vist. Per exemple, una de les capçaleres dels fitxers PE es tracta de les seccions. Les seccions d'un programa són els espais d'adreces diferents en els que s'emmagatzemen les dades seguint una organització concreta. Les quatre seccions més interessants són les següents:

- .text : Conté les instruccions que s'executaran a la CPU. Hauria de ser l'única secció que conté codi.
- .rdata: Normalment conté la informació dels imports i dels exports del programa.
- .data: Conté les dades globals del programa, que és accessible des de tot arreu del mateix programa.
- .rsrc: Aquesta secció conté els recursos fets servir per l'executable que no formen part de l'executable. Normalment és la secció on trobarem més strings i que ens interessa més.

Tenint coneixement a fons de les seccions d'un programa del tipus Portable Executable ens pot donar molta informació. Amb el programa PEview [15] podem navegar per cadascuna d'aquestes seccions i obtenir informació com per exemple quan s'ha generat aquest programa que estem analitzant. A la imatge següent hi ha un exemple d'un anàlisi a un programa fent servir PEview on accedim a les capçaleres per obtenir la data de compilació.



Il·lustració 14 Seccions amb PEview

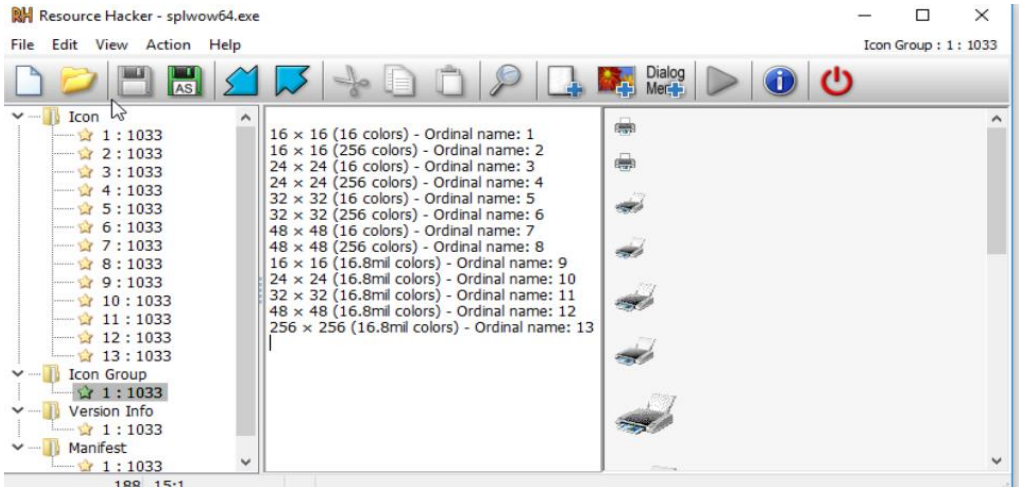
A més a més, podem observar els noms de les seccions ja que es pot modificar al compilar el programa. Si un programa té noms estranys a les seccions, voldrà dir que

està empaquetat i que l'haurem d'executar per analitzar-ho. Cal tenir en compte que aquestes eines no ens permeten concloure si es tracta o no de malware, sinó que cerquem informació per veure si el programa està empaquetat o com s'ha compilat.

Per aconseguir això, també és interessant comprovar les mides de les seccions. Si una secció té la mida virtual molt més gran que les dades en clar, pot fer sospitar de que la majoria de dades es carregaran a memòria durant l'execució i, per tant, estarà empaquetat. Cal tenir en compte que moltes vegades la secció *.data* pot ser més petita en clar que no pas en memòria, ja que això passa a molts programes propietaris de Windows.

Una altra eina que podem fer servir per obtenir més informació del programa de manera estàtica és *Resource Hacker*. Aquest programa recorre la secció *.rsrc* o de recursos i et mostra les imatges, logos, strings, etc. que fa servir el programa.

Per fer una prova he analitzat el programa que gestiona la impressora *splwow64.exe* a la màquina virtual per veure què pot arribar a mostrar.



Il·lustració 15 Exemple execució Resource Hacker.

La imatge anterior és una de les seccions de *.rsrc* on es poden veure logos de la impressora en diferent mida i tipus.

Conclusió anàlisi estàtic

Aquest tipus d'anàlisi ens ha permès obtenir una primera idea del programa que analitzat sense arribar a executar-ho. Obtenint la signatura s'ha pogut comprovar si ja és conegut. A més a més, s'ha pogut extreure els strings del programa per comparar amb diferents IOCs que tinguem a l'abast i que ens seran d'utilitat durant l'anàlisi dinàmic. Un cop realitzat l'anàlisi estàtic podem saber del cert si el programa està empaquetat, el que implica que és necessari seguir amb l'anàlisi dinàmic.

ANALISIS DINÀMIC

L'anàlisi de malware dinàmic és realitza sempre després i durant de l'execució del programa. Normalment s'utilitza en el cas en que l'anàlisi estàtic basic ha arribat a un punt mort i no tenim prou informació com per dir del cert si es tracta de malware o, com bé he comentat, el programa està empaquetat. Aquest tipus d'anàlisi ajuda a veure la funcionalitat del programa, i per això haurem de monitoritzar com afecta aquest al nostre ordinador.

Cal tenir en compte que executar un programa possiblement maliciós pot comprometre el teu dispositiu, tot i que estigui en una màquina virtual. És per això que s'ha de fer amb molt de compte i sempre tenint un entorn controlat en el que les possibilitats de perill siguin les mínimes.

Sandboxes

Per evitar aquest perill moltes vegades es fa servir sandboxes per executar el programa i que retornin un informe del comportament del programa. Una sandbox tracta de protegir el dispositiu d'una manera similar a una màquina virtual, restringint l'espai a disc al que poden accedir i també la memòria disponible. A més a més, no solen tenir xarxa per fer peticions a l'exterior i també es restringeix l'accés a qualsevol altre perifèric. Uns exemples de sandboxes coneguts són els següents:

- Norman Sandbox
- GFI Sandbox
- Joe Sandbox
- Anubis

En aquesta metodologia no ens centrarem en les sandboxes, però si que s'anomenaran els principals desavantatges de les mateixes:

- Simplement executen el programa, no ofereixen la possibilitat d'executar-ho per línia de comandes i per tant no es poden passar paràmetres el que implica que sigui fàcil d'enganyar.
- Si el malware està esperant la comunicació d'un servidor per actuar, no arribarà a executar-se mai o no s'executaran les funcionalitats malicioses.
- Es pot donar el cas en que la sandbox no esperi el temps suficient per a que es donin les funcionalitats o que no es gravin els esdeveniments que ens interessen degut a que no espera suficient.

- El malware pot ser capaç de detectar si s'està executant en una màquina virtual, pel que és necessari realitzar altres tipus d'anàlisis per complementar i per tant no es una solució.
- Si el malware és una llibreria, hi ha funcionalitats que no tenen perquè executar-se i a vegades ni tan sols executa la llibreria.

Aquests inconvenients indiquen que fer servir una sandbox ha de ser un complement a l'anàlisi de malware i en cap cas s'ha de decidir si es tracta de programari lícit només amb l'execució a la sandbox.

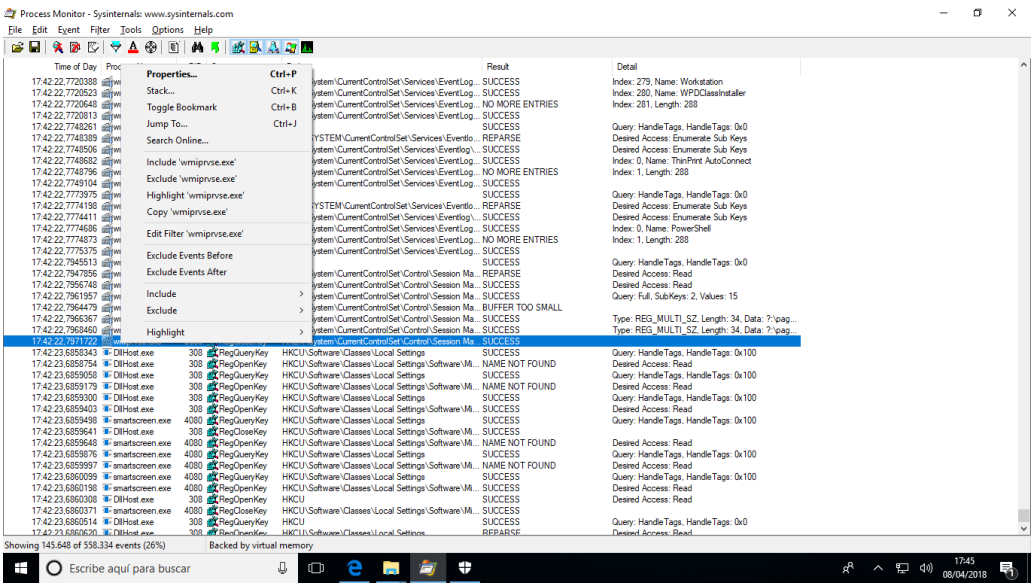
Monitorització de processos amb procmon

La principal eina que farem servir per monitoritzar els processos es tracta de *procmon* [16]. Aquesta eina serveix pel sistema operatiu de Windows i forma part del conjunt d'eines de Microsoft "Windows sysinternals". *Procmon* captura moltes dades com per exemple els canvis a registres, sistema de fitxers, a nivell de xarxa i processos.

Tot i que captura molta informació no ho captura tot, per exemple no captura l'entrada i sortida a dispositius externs igual que no sol capturar les crides a la funció SetWindowsHookEx que serveix per a fer crides a la pantalla (usat per keyloggers), entre d'altres. A més a més, no recomanen utilitzar *procmon* per gravar les comunicacions de xarxa. Per a gravar les comunicacions de xarxa utilitzarem l'eina *Wireshark* explicada més endavant. És per aquests motius que és recomanable acabar l'anàlisi complert i no extreure conclusions d'un anàlisi inacabat, ja que pot ser que estiguem perdent informació que pot ser visible amb altres eines.

Una nota important és que al utilitzar *procmon* es guarden a memòria RAM totes les crides de sistema i les dades que volem, pel que és important no usar *procmon* durant massa estona ja que pot omplir la memòria RAM de la màquina virtual. És per això que quan vulguem capturar haurem de fer click a **"File > Capture events"**. A més a més, també es recomana anar netejant les mostres que tenim fent servir **"Edit > Clear Display"**.

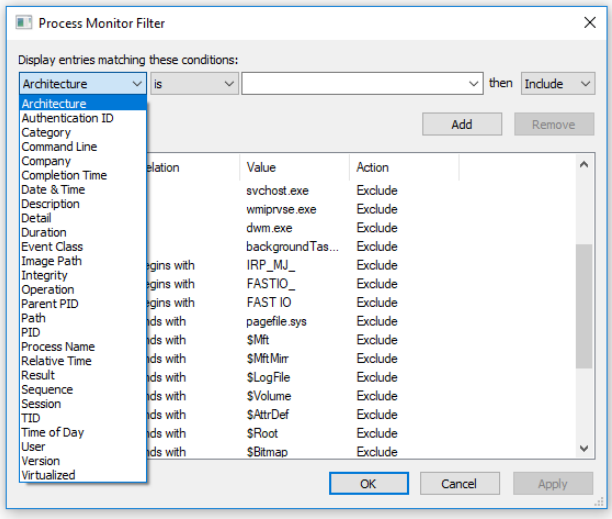
Un cop tinguem tot preparat per a començar l'anàlisi realitzarem els dos passos anteriors i seguidament executarem el malware.



Il·lustració 16 Execució i opcions de procmon

Com podem veure a la imatge anterior, *procmon* ha capturat molts events i sembla força caòtic. La principal gràcia de *procmon* és que podem aplicar filtres per així obtenir les dades que volem. Els filtres són molt útils i hem d'anar observant i recollint dades sobre què realitza el programa en el nostre sistema. Algunes crides a sistema que són interessants filtrar són: RegSetValue, CreateFile, Writefile o ReadFile. És interessant observar si està fent alguna activitat que pugui crear persistència en el sistema, com per exemple crear un procés que s'executarà sempre i que va recollint dades a un fitxer per a, posteriorment, enviar-les a un servidor extern.

Per observar les operacions comentades anteriorment, hem d'aplicar els filtres que hem anomenat abans i anirem a l'apartat de filtres com es pot veure a la imatge següent.



Il·lustració 17 Tipus de filtres de procmon

A més, també podem anar fila per fila excloent els tipus de events que no volguem veure. Ara es tracta d'anar jugant i observant el comportament del programa per concloure si es tracta de malware o no. És important tenir en compte els strings que hem trobat a la primera fase per veure si apareixen novament i poder identificar la seva funció.

Process Explorer

L'eina que es farà servir ara és gratuïta de Microsoft i monitoritza els processos corrent en el sistema. Forma part també del conjunt d'eines de "Windows sysinternals". La manera que té de presentar els processos és molt intuïtiva ja que apareixen en format d'arbre i et permet veure les relacions entre processos, que és important per veure quins processos genera el que s'ha executat.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe		8.728 K	22.916 K	700	Proceso host para los servici...	Microsoft Corporation
ShellExperienceHost....	Susp...	38.012 K	61.760 K	3264	Windows Shell Experience H...	Microsoft Corporation
SearchUI.exe	Susp...	78.572 K	70.560 K	3356	Search and Cortana applicati...	Microsoft Corporation
RuntimeBroker.exe		5.832 K	18.980 K	3444	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		6.436 K	26.852 K	3844	Runtime Broker	Microsoft Corporation
SettingSyncHost.exe		9.452 K	6.428 K	3996	Host Process for Setting Syn...	Microsoft Corporation
MicrosoftEdge.exe		28.628 K	81.748 K	4724	Microsoft Edge	Microsoft Corporation
ApplicationFrameHost...		7.976 K	25.044 K	4732	Application Frame Host	Microsoft Corporation
browser_broker.exe		7.484 K	32.712 K	4808	Browser_Broker	Microsoft Corporation
RuntimeBroker.exe		9.440 K	28.636 K	5092	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		6.096 K	23.688 K	2764	Runtime Broker	Microsoft Corporation
MicrosoftEdgeCP.exe		13.632 K	30.408 K	4372	Microsoft Edge Content Proc...	Microsoft Corporation
MicrosoftEdgeCP.exe		43.696 K	79.940 K	1876	Microsoft Edge Content Proc...	Microsoft Corporation
dllhost.exe		4.548 K	12.076 K	5672	COM Surrogate	Microsoft Corporation
MicrosoftEdgeCP.exe	< 0.01	72.940 K	92.444 K	5948	Microsoft Edge Content Proc...	Microsoft Corporation
RuntimeBroker.exe		1.844 K	11.332 K	2836	Runtime Broker	Microsoft Corporation
MicrosoftEdgeCP.exe	< 0.01	25.096 K	68.680 K	6376	Microsoft Edge Content Proc...	Microsoft Corporation
MicrosoftEdgeCP.exe	< 0.01	27.028 K	90.376 K	2820	Microsoft Edge Content Proc...	Microsoft Corporation
dllhost.exe		2.184 K	13.928 K	5976	COM Surrogate	Microsoft Corporation
SystemSettingsBroker...		2.348 K	13.184 K	7068	System Settings Broker	Microsoft Corporation
SkypeHost.exe	Susp...	22.488 K	1.540 K	6624	Microsoft Skype	Microsoft Corporation
RuntimeBroker.exe		4.360 K	6.708 K	3224	Runtime Broker	Microsoft Corporation
MicrosoftEdgeCP.exe		5.548 K	26.344 K	4268	Microsoft Edge Content Proc...	Microsoft Corporation
MicrosoftEdgeCP.exe		5.500 K	25.936 K	6408	Microsoft Edge Content Proc...	Microsoft Corporation
MicrosoftEdgeCP.exe	< 0.01	50.580 K	103.052 K	2628	Microsoft Edge Content Proc...	Microsoft Corporation
svchost.exe		6.468 K	12.224 K	752	Proceso host para los servici...	Microsoft Corporation
svchost.exe		19.516 K	42.856 K	916	Proceso host para los servici...	Microsoft Corporation
sihost.exe		6.940 K	26.720 K	2876	Shell Infrastructure Host	Microsoft Corporation
taskhostw.exe		8.408 K	18.576 K	2244	Proceso de host para tareas ...	Microsoft Corporation

Il·lustració 18 Exemple d'execució de Process Explorer

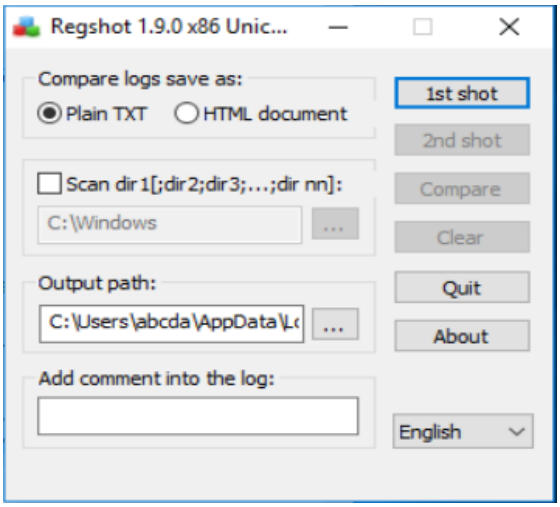
Aquest programa mostra el identificador del process, l'ús de CPU, etc. La vista s'actualitza cada segon. És important saber que per defecte els serveis estan marcats de color rosa, els processos de color blau, els processos nous de color verd i els que finalitzen de color vermell.

És molt fàcil identificar documents del tipus PDF o Microsoft Word maliciosos fent servir *process explorer* ja que si una macro executa una comanda per arribar a la terminal, es veu perfectament amb aquest programa.

Per agilitzar aquesta part, farem servir també el programa *Regshot* [17], que realitza com una imatge de l'estat dels registres i fa la comparació amb una altra imatge. És una eina molt útil per obtenir tota la informació dels registres.

Regshot

Com he comentat, primer de tot s'ha de realitzar una imatge dels registres fent click al botó "1st shot" que podem observar a la imatge de *Regshot* següent:



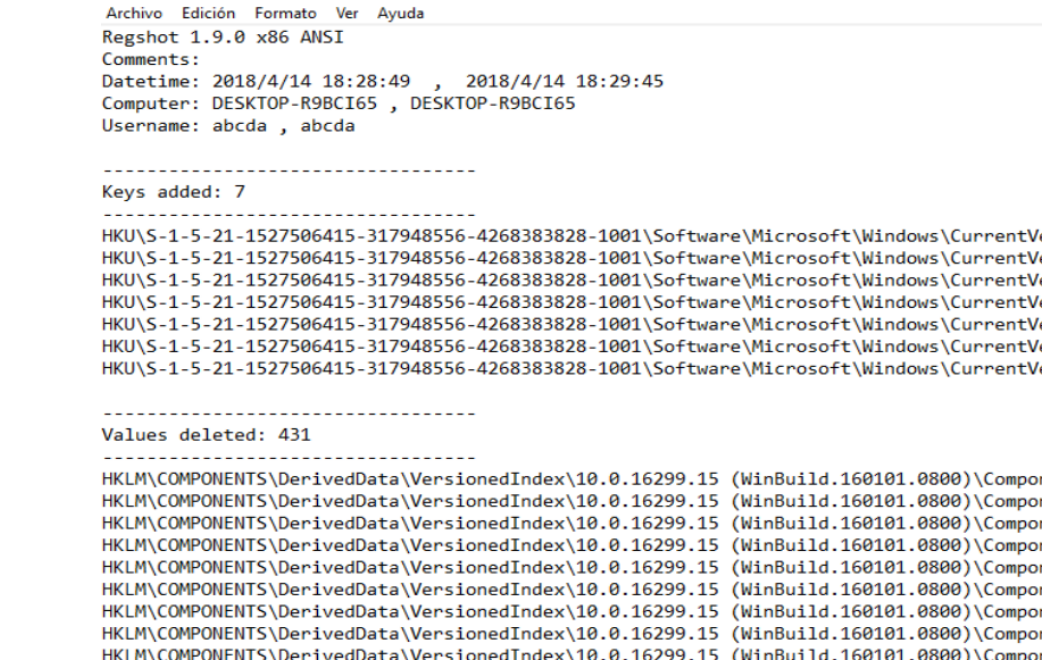
Il·lustració 19 Programa Regshot

Quan hem emmagatzemat la primera imatge i tenim el *process explorer* monitoritzant els processos, serà el moment d'executar el que creiem que es tracta de malware i observem el *process explorer* per veure si genera nous processos.

Un cop hem deixat funcionar el malware una estona i sense haver realitzat més accions al dispositiu, procedim a realitzar la segona imatge dels registres. Cal tenir en compte que hi ha programes maliciosos que el que fan és cridar a la funció "sleep", que simplement espera durant una estona per així evitar que el detectin quan l'executen. És per això, que sempre és recomanable deixar una bona estona el malware funcionant ja que sinó podem creure que es tracta de programari lícit de manera equivocada.

La segona imatge la farem quan creguem que el malware ja ha fet la seva funció i ha infectat el nostre sistema. És llavors quan des del mateix *Regshot* procedim a comparar els les dos imatges que hem realitzat i ens generarà un fitxer amb el resultat.

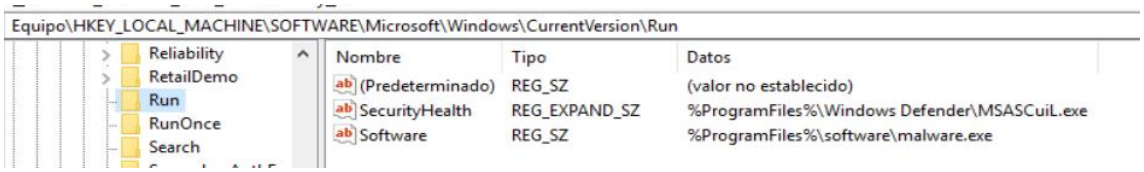
La comparació dels registres ens permetrà saber si ha realitzat canvis que pretenguin ser persistents en el sistema. A la següent imatge podem trobar una comparació de dues imatges realitzades amb *Regshot* en que el fitxer resultant apunta totes les modificacions realitzades als registres.



Il·lustració 20 Exemple resultat Regshot

Tot i que també podem trobar molt soroll que simplement pot intentar amagar la verdadera funció. El més important és tenir paciència i dedicar-se a anar descartant el que no ens és útil. A més a més, no ens hem de centrar en una cosa concreta ja que ens podem perdre dades molt més interessants que poden ser més decisives.

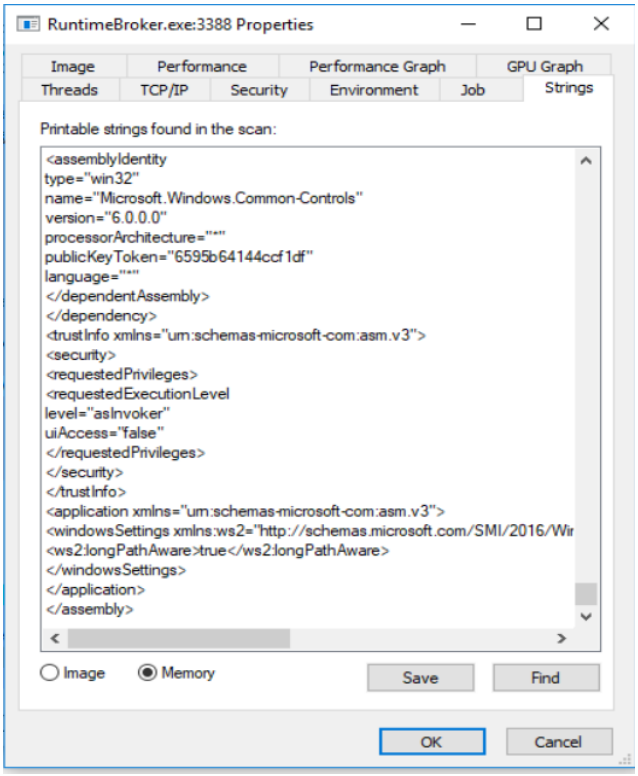
Un cop tenim la comparació dels registres, podem cercar quins canvis s'han realitzat a partir de la utilitat de cerca per defecte que ve amb el bloc de notes. La cerca la farem de la informació obtinguda fins ara. Com a exemple, si fent servir l'eina *procmon* veiem que s'ha generat un fitxer, podem cercar per aquest i veure si s'ha afegit un valor a un registre que s'inicia sempre que arrenca el sistema. A la següent imatge hi ha un exemple d'aquest cas:



Il·lustració 21 Exemple canvi persistent a un registre

Tornant al *process explorer* que hem obert paral·lelament, es pot veure si ha generat nous processos a partir dels colors que hem comentat. Per cada procés generat i quan està el malware funcionant, podem observar quines llibreries ha importat a memòria. És interessant recórrer tots els apartats que podem veure a la següent imatge per tal d'obtenir el màxim d'informació de cada procés.

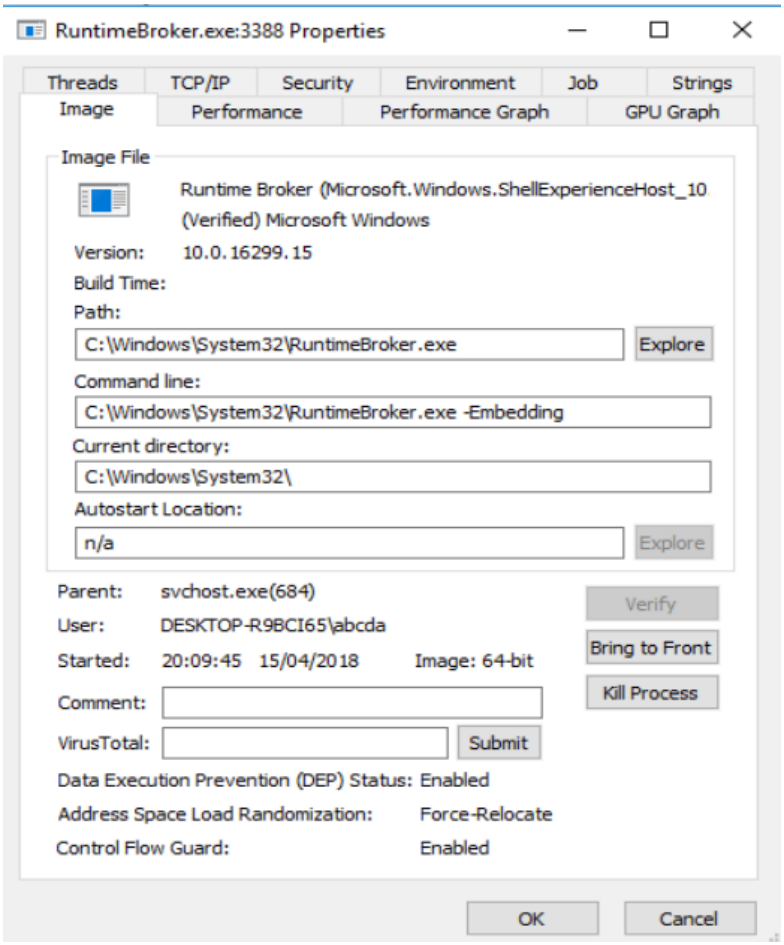
Una d'aquestes és la de "Strings", que et permet veure els strings en memòria. Aquests strings, ens haguessin passat desapercebuts si el malware estava empaquetat, ja que fins que no es carrega tot el contingut en memòria per a l'execució, no podem tenir accés.



Il·lustració 22 Opcions del Process Explorer

Una altra opció interessant és la de verificar l'origen del programa que es troba a l'apartat "Image". Aquesta opció compara la signatura digital del programa amb la oficial de Microsoft per verificar que és legítim. És útil perquè el malware molt sovint modifica els fitxers de Windows per introduir el seu codi.

Seguidament tenim un exemple d'un programa verificat de Microsoft. Com es pot veure, just a sota del nom del fitxer trobem "(Verified)" que ha aparegut després de fer click al botó "Verify" que apareix a la dreta de la imatge.

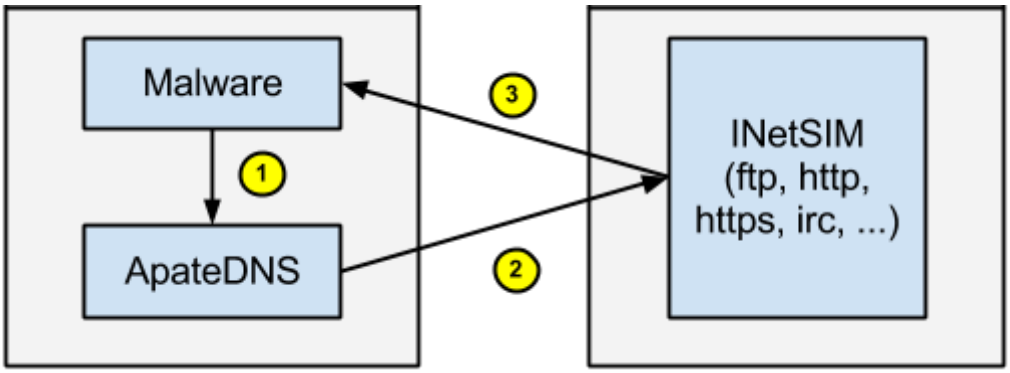


Il·lustració 23 Opcions del programa Process Explorer 2

Ara només ens queda seguir investigant per totes les pestanyes i obtenir el màxim d'informació del programa.

Analitzant la xarxa

Una part que no hem cobert encara es la de les comunicacions que realitza un programa a nivell de xarxa. És molt interessant tenir en compte aquest aspecte ja que molts cops els programes maliciosos es manipulen des de servidors coneguts com "Command and Control" (C&C). Com que no podem donar accés a través del nostre ordinador cap a fora per seguretat i privacitat ja que no volem que obtinguin la nostra IP, el que farem serà crear una xarxa de mentida que simularà tots els ports típics que es fan servir, i analitzarem amb WireShark les comunicacions sortint des de la pròpia màquina. Per analitzar la xarxa seguirem el següent esquema:

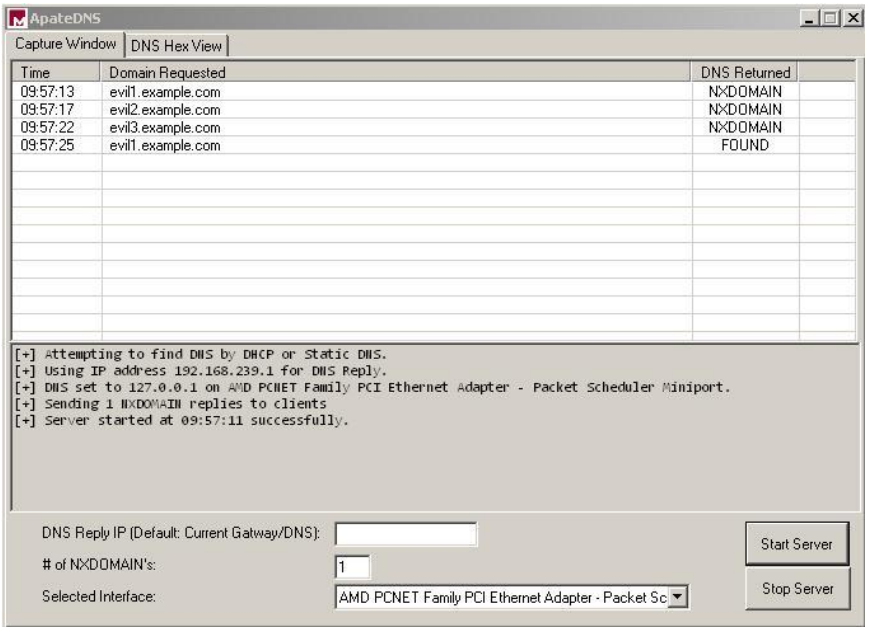


Il·lustració 24 Esquema laboratori

INetSim [18] es l'eina que farem servir per simular que un programa es comunica amb l'exterior i tindrà tots els ports comuns oberts. Per a assolir això, haurem d'instal·lar el programa anterior a la segona màquina virtual que hem fet al principi, amb el sistema operatiu Ubuntu. A més a més, haurem d'instal·lar ApateDNS per redirigir el tràfic a la màquina virtual que servirà com a servidor. ApateDNS representarà el DNS del dispositiu on es realitzen les proves, i que redirigirà totes les comunicacions cap al servidor d'INetSim.

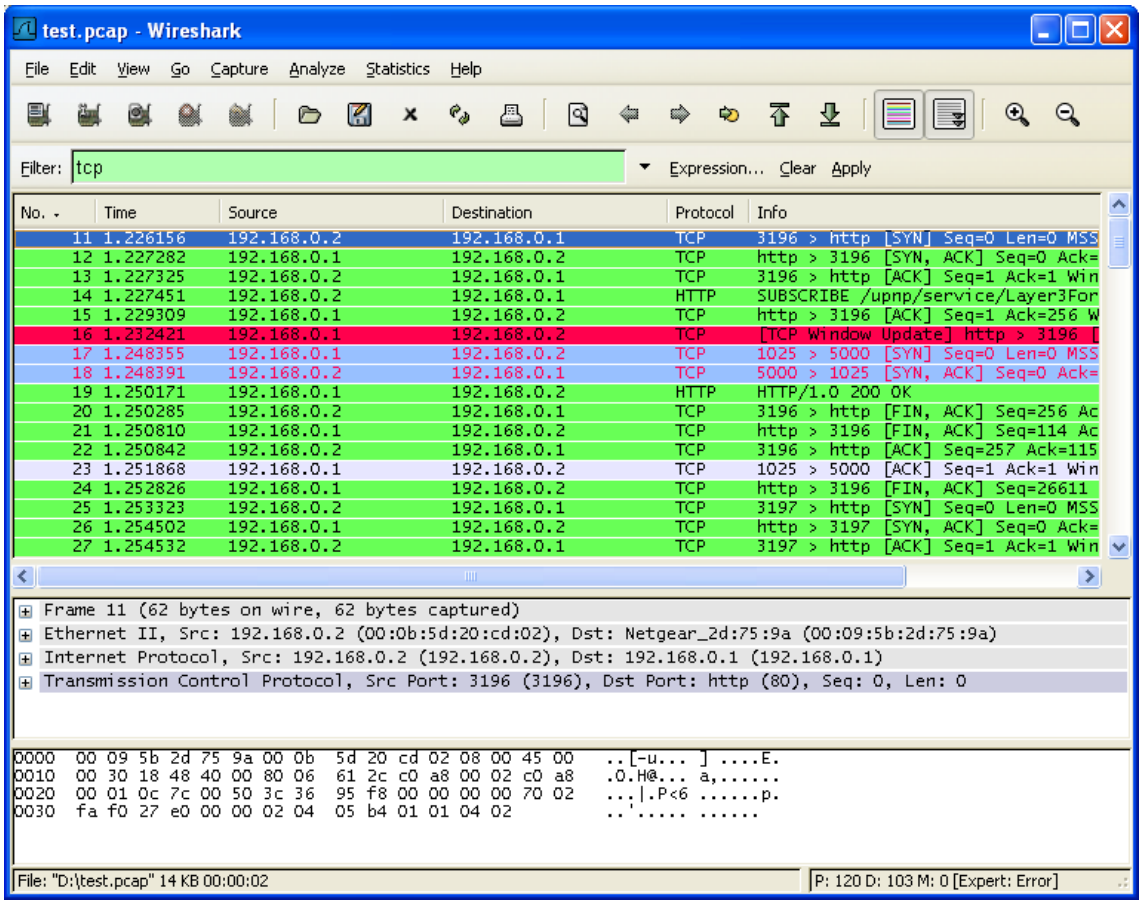
Un cop tenim la màquina virtual ubuntu amb el INetSIM engegat i l'ApateDNS donant servei com a DNS, procedim a executar el Wireshark per a gravar les comunicacions sortints. Wireshark és un analitzador de paquets opensource que monitoritza totes les comunicacions de manera similar a l'eina "tcpdump" nativa en Linux, tot i que fa servir una interfície gràfica molt més amigable.

Quan està tot preparat, és el moment d'executar el malware. Després de deixar una estona en funcionament igual que abans, observem els events que han enregistrat les eines, com a la següent imatge:



Il·lustració 25 Exemple de resultat a ApateDNS

Podem observar que el ApateDNS té enregistrades 4 preguntes que ha realitzat el malware al DNS i a quin hostname ha realitzat la pregunta. A més a més, personalment trobo que s'ha de inspeccionar a fons el registre de Wireshark ja que queda molt detallada la informació. Amb wireshark, igual que amb procmon, podem aplicar filtres per veure els tipus de paquets enviats o rebuts, així com el protocol fet servir o el destí dels paquets. Podem aprofitar tota la informació anterior per filtrar respecte aquesta com, per exemple, dominis que haguem vist a l'apartat d'strings. A la següent imatge s'hi ha aplicat un filtre per paquets que utilitzen el protocol TCP:



Il·lustració 26 Exemple de filtre a Wireshark

Wireshark és una eina que té un gran nombre de funcionalitats, com per exemple cercar els dominis que s'han demanat que es resolguin, els resolts, les dades de les comunicacions, etc. Donada la gran capacitat de filtres que ofereix i la dificultat d'analitzar fitxers del tipus *pcap*, a la bibliografia deixo un enllaç amb diferents pràctiques per analitzar comunicacions a la xarxa [19].

Conclusió anàlisi dinàmic

Obtenir informació de totes aquestes dades per extreure una conclusió és el nostre objectiu, i per això cal donar-hi moltes voltes i inspeccionar tot el que puguem.

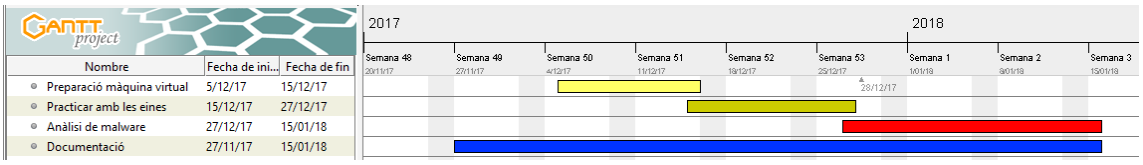
Un cop obtinguda tota la informació possible del subjecte analitzat, s'ha de concloure sobre si es tracta d'un programa maliciós i actuar conseqüentment.

PLANIFICACIÓ

La planificació d'aquest projecte es divideix en quatre grans fases. Aquestes quatre fases són:

- Entorn de laboratori
- Eines per a fer l'anàlisi
- Anàlisi de malware pràctic
- Documentació de tot el projecte

En el següent diagrama de Gantt es pot veure la planificació inicial del projecte.

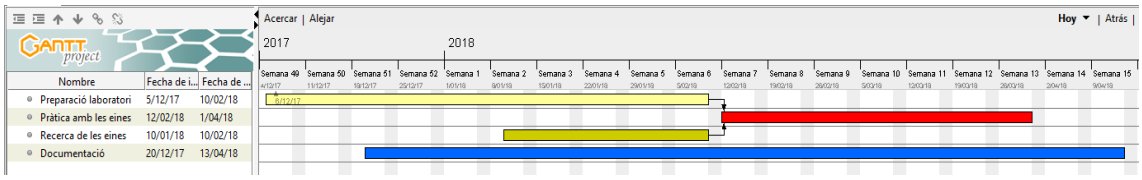


Il·lustració 27 Diagrama de Gantt antic

Cadascuna d'aquestes quatre fases és important pel desenvolupament del projecte. Les tres primeres fases tenen més relació amb la part tècnica, i també estan totalment relacionades entre elles temporalment ja que per avançar a la següent primer s'ha d'acabar l'anterior. La fase de documentació en canvi, és transversal durant tot el projecte i tracta d'anar recollint i redactant tot el progrés que es fa així com el context i l'estat de l'art.

Seguint la metodologia del projecte es pot veure clarament l'encaix de les tres fases tècniques dins del procés. En la planificació inicial del projecte, la fase de l'entorn de laboratori té relació amb la tasca de crear la màquina virtual i afegir-hi tot aquell programari que creiem que pugui arribar a explotar un malware. Tot i això, un cop realitzat el projecte he comprovat que s'ha emmarcat en el mateix temps que la fase de recerca de les eines per a fer l'anàlisi. És per això que a la planificació final que es pot observar més endavant, he modificat el nom de la fase "practicar amb les eines" a "recerca de les eines". Donat que practicar amb les eines a la pràctica ha sigut com realitzar anàlisis de malware, la fase "Anàlisi de malware" l'he anomenat "practicar amb les eines" ja que així segueix el fil conductor del projecte.

Amb les modificacions anteriors també dono cabuda a la feina de recerca realitzada en el projecte, que inclou les definicions de malware, recerca d'eines d'anàlisi, etc.



Il·lustració 28 Diagrama de Gantt final

A la il·lustració anterior es pot veure quina ha sigut la planificació final, i com estan relacionades entre si les fases explicades. Com he comentat abans, la fase de preparació de l'entorn i de recerca de les eines han compartit espai temporal ja que la recerca de les eines significava modificar el laboratori per adaptar-ho a les necessitats de l'anàlisi.

Com bé està indicat, les fases anteriors són necessàries per al desenvolupament de la tercera fase, la de la pràctica amb les eines. Aquesta fase tracta de posar en pràctica tots els coneixements obtinguts fent servir les eines i l'entorn preparat.

Per acabar, la fase de documentació s'ha anat realitzant al llarg del projecte ja que s'ha anat escrivint i emmagatzemant imatges durant el procés.

IMPACTE E INFORME DE SOSTENIBILITAT

A l'hora de plantejar l'impacte que té aquest projecte sobre l'economia, la societat i el medi ambient he fet servir la matriu de sostenibilitat que ofereix la FIB als estudiants.

	PPP	Vida Útil	Riesgos
Ambiental	Consumo de diseño	Huella ecológica	Ambientales
Económico	Factura	Plan de viabilidad	Económicos
Social	Impacto personal	Impacto social	Sociales

Taula 1 Informe de sostenibilitat

Seguidament, deixo una explicació dels plantejaments de cada cel·la de la taula anterior.

Ambiental / PPP: l'impacte ambiental de la posta a punt del projecte és molt baixa donat que no es necessiten més recursos que la llum necessària per mantenir el dispositiu engegat. No hi ha residus al implantar el projecte.

Ambiental / Vida útil: Igual que en el cas anterior, no es generen residus que impactin la petjada ecològica. Si que es necessita energia elèctrica per fer funcionar l'ordinador.

Ambiental / Riscs: El principal risc és que el dispositiu que es fa servir per analitzar el malware s'espatlli i que, per tant, s'hagi de reemplaçar amb un altre dispositiu nou. Això genera un impacte a la petjada ecològica i s'intenta evitar congelant imatges que es tornaran a carregar després de cada anàlisi per a que una infecció no pugui fer malbé el dispositiu sobrecarregant la màquina.

Econòmic / PPP: Els costos del projectes són principalment humans donades les hores que requereix fer recerca sobre el sector i varia en funció del nombre d'anàlisis que es demanin, ja que se sol contractar per a fer una gran quantitat d'anàlisis i no per un de sol. A més a més, també cal tenir en compte el dispositiu on es realitzaran aquests anàlisis, que en funció de la potencia serà més o menys car però s'ha de tenir en compte que requereix de potencia suficient com per mantenir dues màquines virtuals de manera fluida.

Econòmic / Vida útil: La vida útil del projecte un cop implantat s'espera que sigui de mínim 10 anys tot i que s'ha de tenir en compte que la tecnologia avança molt ràpidament i potser és necessari adaptar-se a noves eines. Durant tot aquest temps es van realitzant anàlisis que, a un cost molt baix, es pot cobrar per paquets o unitats que faran que sigui rentable.

Econòmic / Riscs: Un possible risc és que no sigui comprat ni llogat per ningú i, per tant, no serà rentable.

Social / PPP: El principal impacte social que ha tingut aquest projecte en mi és la consciencia i el coneixement adquirit sobre el malware, a més a més de la identificació del mateix per tal de protegir-me.

Social / Vida útil: Penso que el projecte pot apropar la seguretat a moltes persones i m'agradaria que ajudés a conscienciar sobre la vulnerabilitat que patim a la xarxa.

Social / Riscs: Un risc que pot haver-hi és si apareix un nou malware que és capaç de passar per alt qualsevol d'aquestes eines i acabar infectant un dispositiu sense que l'usuari se n'adoni. És poc probable però pot anar lligat amb els avenços de la tecnologia.

PRESSUPOST

Donat que aquest projecte intenta fer servir programes Open Source i que el hardware necessari per a fer funcionar els programes requerits és relativament baix, el cost final de projecte s'estima que serà molt assequible.

Pel que he comentat abans, el principal cost del projecte serà humà. Cal tenir en compte que bona part del projecte tracta d'investigar quin és l'estat de l'art dels programes maliciosos, així com les eines que es fan servir per analitzar-los. És per això que el cost humà relacionat amb el projecte és gran i varia en funció del nombre d'anàlisis que es realitzen.

El cost del projecte s'ha analitzant en el seu conjunt i no s'ha desglossat per apartat ja que s'entén que l'objectiu d'aplicar el projecte és arribar a una conclusió que només ens és vàlida en el cas en que s'ha seguit tot el procediment.

En quant al cos humà relacionat amb el projecte, s'ha de mesurar relativament a la disponibilitat de la documentació i la disponibilitat d'autogestionar-se per a implementar aquest projecte en un entorn propi. És a dir que, com que tota la informació està disponible al públic, deixa de ser limitada i per tant simplement s'ha de valorar l'aplicació i estat de l'art d'aquesta i no tant en clau de I+D ja que no forma part del nostre objectiu.

He valorat els costos directes com a l'ordinador físic on es realitzaran les proves, així com el conjunt de perifèrics per a una millor experiència. Donat els requeriments de potència he valorat que l'ordinador ha de tenir un cost d'uns 600€ a 800€. Cal tenir en compte que, com en el meu cas, se solen reutilitzar ordinadors que ja no donen tanta utilitat per a realitzar aquests tipus de projectes, per tant el cost directe del projecte es veuria reduït a zero un cop es reutilitzen també els perifèrics.

Com a resum he fet una taula que exposa tots els comentaris anteriors.

Activitat	Import	Observacions
Estudi de l'estat de l'art	250€ / mes	He calculat unes 25 hores al mes a 10€/hora
Aprendre les eines de la metodologia	500 €	He valorat els coneixements que són públics per tothom
Ordinador laboratori	700 €	
Cost per cada anàlisi	50 €	Aquest preu és orientatiu.
TOTAL CD	1200€ + 250€/mes	Cal sumar el cost variable per nombre d'anàlisis
Llum	50€/mes	Preu orientatiu
TOTAL CD + CDI	1200€ + 300€/mes	Cal sumar el cost variable per nombre d'anàlisis
Ordinador de contingència	700 €	En cas de necessitat degut a que s'hagi espatllat l'ordinador que es fa servir com a laboratori
TOTAL CD + CDI + Cont.	1900€ + 300€/mes	Cal sumar el cost variable per nombre d'anàlisis
TOTAL	1900€+300€/ mes + 50€xN	Sent N el nombre d'anàlisis en un mes

Taula 2 Pressupost

CONCLUSIONS

En aquest projecte s'ha donat un conjunt de directrius per a realitzar un anàlisi de malware de manera mínimament ràpida i eficaç. A més a més, s'ha posat en context quina és la situació respecte el software maliciós arreu del món i quins són els mètodes fets servir fins ara per a protegir-se. Les directrius comprenen des de la preparació del laboratori a l'anàlisi estàtic i dinàmic.

A la preparació del laboratori s'han fet servir dues màquines virtuals per a seguretat del sistema i la facilitat de poder tornar a l'estat abans de la infecció. A més a més, s'ha recomanat la instal·lació d'un conjunt d'eines que es faran servir durant l'anàlisi per a que s'instal·lin durant la preparació.

Fent servir les eines comentades abans, en aquest projecte s'explica com obtenir una signatura única d'un fitxer i comparar-ho amb bases de dades conegudes fent servir *md5deep*, la manera de llistar les cadenes de caràcters d'un programa amb el software *strings* i *radare2* i com explorar les seccions i els recursos del software sense necessitat d'executar-ho amb les eines *PEview* i *Resource Hacker*.

Per acabar, s'han fet servir eines com *procmon* i *process explorer* per a la monitorització de procesos, l'eina *Regshot* per comparar els canvis realitzats als registres en un dispositiu entre dues imatges fetes per nosaltres i finalment, s'ha analitzat les comunicacions a la xarxa realitzades per el software analitzat cercant indicis de comportament del programa amb les eines *Wireshark, INetSim* i *ApateDNS*.

PROPOSTES DE MILLORA

Aquest projecte és molt ampli i tracta moltes eines diferents amb utilitats molt diferents entre elles tot i estar relacionades dins del sector de l'anàlisi de dispositius. Al tenir tantes eines diferents no he pogut profunditzar en cap en concret tot i que si que he pogut investigar sobre les funcionalitats del W*ireshark* tot i no estar reflexat en aquest projecte.

M'hagués agradat poder haver tingut més temps per profunditzar més en el projecte sobre aquelles eines que tenen més funcionalitats com per exemple *Wireshark*, *radare2*, *procmon* i descobrir de noves que donat l'abast del projecte m'ha sigut impossible afegir-les.

A més a més, m'hagués agradat explorar sobre el món del *reversing*, que tracta de entendre el funcionament d'un programa fent servir eines com *radare2* o *Frida.*

BIBLIOGRAFIA

Context

[1] Jonathan Lemonnier, “What is malware?”, 2016 [Online]:
https://www.avg.com/en/signal/what-is-malware

[2] Neil Dupaul, “Common Malware Types” [Online]:
https://www.veracode.com/blog/2012/10/common-malware-types-cybersecurity-101

[3] AV-test, “Malware statistics and trends” [Online]:
https://www.av-test.org/en/statistics/malware/

[4] Malwarebytes, “Mallwarebytes annual report”, 2018 [Online]:
https://press.malwarebytes.com/2018/01/25/malwarebytes-annual-state-malware-report-reveals-ransomware-detections-increased-90-percent/

Preparació de l’entorn

[5] Descàrrega i documentació de virtualbox [Online]:
https://www.virtualbox.org/

[6] Descàrrega i documentació de Ubuntu [Online]:
https://www.ubuntu.com/

[7] Descàrrega de *procmon* [Online]:
https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

[8] Descàrrega de *regshot* [Online]:
https://sourceforge.net/projects/regshot/

[9] Descàrrega de *strings* [Online]:
https://docs.microsoft.com/en-us/sysinternals/downloads/strings

[10] Descàrrega de *inetsim* [Online]:
http://www.inetsim.org/

[11] Descàrrega de *ApateDNS* [Online]:
https://www.fireeye.com/services/freeware/apatedns.html

[12] Descàrrega de *Wireshark* [Online]:
https://www.wireshark.org/

[13] Descàrrega de *PEview* [Online]:
https://sourceforge.net/projects/peview/

[14] Michael Sikorski and Andrew Honig, “Practical Malware Analysis”, 2012 [Llibre]

Enllaços d'interès per a la realització de l'anàlisi

[15] Malwarebytes, "Five PE Analysis Tools Worth Looking At", 2014 [Online]: https://blog.malwarebytes.com/threat-analysis/2014/05/five-pe-analysis-tools-worth-looking-at/

[16] Microsoft - J.C. Hornbeck, "Process Monitor – Hands-On Labs and Examples", 2008 [Online]: https://blogs.technet.microsoft.com/appv/2008/01/24/process-monitor-hands-on-labs-and-examples/

[17] Martin Hendrikx, "How to Use Regshot To Monitor Your Registry", 2014 [Online]: https://www.howtogeek.com/198679/how-to-use-regshot-to-monitor-your-registry/

[18] Tech Anarchy, "Installing and Configuring InetSim", 2013 [Online]: https://techanarchy.net/2013/08/installing-and-configuring-inetsim/

[19] Malware-Traffic-Analysis, pràctiques d'anàlisis de *pcaps* [Online]: https://www.malware-traffic-analysis.net/